In the Claims:

This listing of claims replaces all prior versions and listings of claims in the application.

1. (Currently amended) A semiconductor integrated circuit ~~arranged~~ to execute application code to be received from a memory via external connections, comprising:

a processor ~~for executing~~ to execute the application code from the memory;

an internal bus ~~within the integrated circuit for providing~~ to provide the application code to the processor from the memory ~~external connections~~;

a verifier processor ~~arranged~~ to receive the application code via the internal bus, wherein the verifier processor ~~is arranged to~~ continually processes ~~process~~ the application code using a verification function while ~~whilst~~ the processor executes the application code from the memory independently of the verifier processor, and to impair the function of the integrated circuit in ~~the~~ an event that the application code does not satisfy the verification function; ~~,~~ and

an instruction monitor ~~arranged~~ to monitor code requests issued by the processor and to impair the function of the integrated circuit unless ~~the address~~ addresses of the code requests fall ~~falls~~ within a given range.

2. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 further comprising an internal

memory; wherein the given range is ~~predefined and~~ stored in the ~~an~~ internal memory.

3. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 wherein the given range is derived by the verifier processor during a first check of the memory.

4. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 3 wherein the application code in memory comprises ~~is in the form of~~ a linked list; and wherein ~~and~~ the given range comprises a table of linked list addresses.

5. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 3 wherein the verifier processor is ~~arranged~~ to impair the function of the integrated circuit if the verification function is not completed for one complete cycle of the linked list within a predetermined time.

6. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 wherein the verifier processor is ~~arranged~~ to receive pause and stop requests and is configured so that any pause and stop request is ineffective during a first check of the code.

7. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 wherein the verifier processor ~~can only be~~ is paused for only a predetermined time.

8. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 wherein if the application codes does not satisfy the verification function, a reset signal is asserted after a predetermined time.

9. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 8 wherein a status signal is set and stored to indicate that the code does not satisfy the verification function before the reset signal is asserted.

10. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 wherein the verification function includes a hash function on the application code.

11. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 wherein the verifier processor is ~~arranged~~ to receive a stored secret from the memory; and wherein ~~and~~ the verification function comprises ~~is~~ a comparison of the secret and the processed application code.

12. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 wherein the verification function comprises:

hashing the application code to produce hashed code; ~~r~~

retrieving a signature of the application code

from a signature store within the memory; and

verifying the hashed code and the signature using a public key.

13. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 wherein the verifier processor comprises ~~has~~ a stop input; and ~~is arranged to restart~~ wherein the verifier processor is to restart a given time period after a stop [[,]] and ~~arranged~~ does not ~~to~~ stop again until completing the verification function on the application code at least once.

14. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 wherein the verifier processor is to request ~~requests~~ portions of the application code from the ~~flash~~ memory at intervals between requests by the processor for portions of the application code.

15. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 14 wherein the verifier processor is to request ~~requests~~ portions of application code at less frequent intervals than the processor.

16. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 14 wherein the verifier processor is ~~arranged~~ to request portions of the application code at pseudo random times.

17. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 14 wherein the verifier processor is ~~arranged~~ to carry out read requests at a faster rate during a first check than in subsequent checks.

18. (Currently amended) ~~A~~ The semiconductor integrated circuit according to claim 1 wherein impairing the function of the integrated circuit comprises resetting the integrated circuit.

19. (Currently amended) A semiconductor integrated circuit ~~arranged~~ to execute application code to be received from ~~an external~~ a memory ~~via an external connection~~, comprising:

a processor ~~for executing~~ to execute the application code from the memory;

an internal bus ~~within the integrated circuit and~~ connected to the processor to provide the application code to the processor from the memory; ~~external connections; and~~

a verifier processor ~~arranged~~ to receive the application code via the internal bus, wherein the verifier processor ~~is structured to process~~ processes the application code using a verification function while the processor executes the application code from the memory independently of the verifier processor, and to impair ~~impede~~ the execution of the integrated circuit if the application code does not satisfy the verification function; and

an instruction monitor to be connected to the

internal bus, to monitor code requests issued by the processor, and to impair the execution of the integrated circuit unless addresses of the code requests fall within a given range.

20. (Canceled).

21. (Currently amended) The semiconductor integrated circuit of claim 19 ~~claim 20~~ wherein the given range is derived by the verifier processor during a check of the memory.

22. (Currently amended) The semiconductor integrated circuit of claim 19 ~~claim 20~~ wherein the application code in memory comprises ~~is accessed by~~ a linked list; and wherein the given range is stored in a table of linked list addresses.

23. (Currently amended) The semiconductor integrated circuit of claim 19 wherein the verification processor is ~~structured~~ to impair the execution of the integrated circuit by asserting a reset signal to the processor if the application ~~codes~~ code does not satisfy the verification function within a predetermined time.

24. (Currently amended) The semiconductor integrated circuit of claim 19 wherein the verification processor includes:

an internal processor to coordinate ~~that coordinates the~~ processing of the application code using the verification function and ~~impairs~~ to impair the execution of the integrated circuit if the application code does not satisfy the verification function;

a code memory [[,]] to be coupled to the internal processor, ~~that stores~~ to store code for controlling the internal processor to process the application code, and to impair the execution of the integrated circuit if the application code does not satisfy the verification function; and

an interface circuit to be connected to ~~that connects~~ the internal processor with the internal bus.

25. (Currently amended) A memory system, comprising:

a non-volatile memory ~~that~~ to store ~~stores~~ application code; and

a semiconductor integrated circuit ~~arranged~~ to execute the application code to be received from the non-volatile memory ~~via an external connection~~, the integrated circuit including[[:]]

a processor to execute ~~for executing~~ the application code from the non-volatile memory,~~;~~

an internal bus ~~within the integrated circuit and~~ connected to the processor to provide the application code to the processor from the non-volatile memory; ~~external connections; and~~

a verifier processor ~~arranged~~ to receive the application code via the internal bus, wherein the verifier processor ~~is structured to process~~ processes the application code using a verification function while the processor executes the application code from the non-volatile memory independently of the verifier processor, and to render the memory system ~~wholly or~~ at least partly unusable if the application code does not satisfy the verification function, and

an instruction monitor to be connected to the internal bus, to monitor code requests issued by the processor, and to impair the execution of the integrated circuit unless addresses of the code requests fall within a given range.

26. (Canceled).

27. (Currently amended) The memory system of claim 25 ~~claim 26~~ wherein the given range is derived by the verifier processor during a check of the non-volatile memory.

28. (Currently amended) The memory system of claim 25 ~~claim 26~~ further comprising an internal memory; wherein the non-volatile memory includes a linked list for accessing the application code; and wherein the given range is stored in the ~~an~~ internal memory of the integrated circuit as a table of linked list addresses.

29. (Currently amended) The memory system of claim 25 wherein the verification processor is ~~structured~~ to impair the execution of the integrated circuit by asserting a reset signal to the processor if the application code ~~codes~~ does not satisfy the verification function within a predetermined time.

30. (Currently amended) The memory system of claim 25 wherein the verification processor includes:

an internal processor ~~that~~ to coordinate ~~coordinates~~ the processing of the application code using the verification function and to impair ~~impairs~~ the execution of the integrated circuit if the application code does not satisfy the verification function;

a code memory [[,]] to be coupled to the internal processor, ~~that stores~~ to store code for controlling the internal processor to process the application code, and to impair the execution of the integrated circuit if the application code does not satisfy the verification function; and

an interface circuit to be connected to ~~that connects~~ the internal processor with the internal bus.

31. (New) A method for executing application code received from an external memory via external connections, the method comprising:

executing application code from the external memory with a processor;

providing the application code to the processor via

an internal bus;

providing the application code to a verifier processor via the internal bus;

continually processing the application code with the verifier processor, while the processor executes the application code independently of the verifier processor, using a verification function;

monitoring code requests issued by the processor with an instruction monitor; and

impairing operation of the integrated circuit if the application code does not satisfy the verification function or if addresses of the code requests fall outside a given range.

32. (New) The method of claim 31 further comprising deriving the given range with the verifier processor during a check of the external memory.

33. (New) The method of claim 31 further comprising storing the given range in an internal memory.

34. (New) The method of claim 31 further comprising:

receiving pause and stop requests at the verifier processor; and

configuring the verifier processor so that any pause and stop request is ineffective during a first check of the code.